NISTIR 4650

Guidelines for the Development of Computer Based Models in a Cementitious Materials Modeling Laboratory

Lawrence J. Kaetzel
James R. Clifton
Leslie J. Struble*

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Building and Fire Research Laboratory
Gaithersburg, MD 20899

*Civil Engineering Department
University of Illinois
Urbana, Illinois 61801

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

NIST

NISTIR 4650

Guidelines for the Development of Computer Based Models in a Cementitious Materials Modeling Laboratory

Lawrence J. Kaetzel
James R. Clifton
Leslie J. Struble*

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Building and Fire Research Laboratory
Gaithersburg, MD 20899

*Civil Engineering Department
University of Illinois
Urbana, Illinois 61801

August 1991

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

ABSTRACT

This paper presents guidelines and considerations for the development, archiving, and distribution of computer models for a centralized cementitious materials modeling laboratory. An analysis of the approach used by cement researchers to develop large complex computer models reveals the need for guidelines in selecting computer platforms, software languages and tools, software engineering and documentation. A modeling laboratory established at NIST associated with the NSF Center for Science and Technology of Advanced Cement-Based Materials is discussed and the important of such a facility in promoting the exchange of information (i.e. ideas, models, data). The computer models currently archived in the modeling laboratory are use as test cases to describe the facility.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Computer based modeling is an important component in the development of a knowledge system for cement and concrete research (Figure 1). However, to realize the value of computer modeling efforts, considerations must be made for development, preservation, and the dissemination of models to the cement and concrete research community.

Developmental approaches to computer based modeling have been previously documented in texts and technical papers [Spriet 82, Cross 79], yet there is no clear-cut technique for building mathematical models for a broad spectrum of real-world processes. The approach taken by cement and concrete researchers is often influenced by several factors:

[1] previous work by colleagues in a related area
[2] the available human resources
[3] the available computer resources

This paper addresses issues that relate to the development of computer based models: specifically, the design of computer programs, access to models, documentation, and archive and distribution issues are addressed. It is written to assist current and potential computer modelers in the development of cementitious materials models, so that models can be preserved and further enhancements can be made.

Throughout this paper the activities of the Cementitious Materials Modeling Laboratory (CMML) that is currently operating at the National Institute of Standards and Technology, Building Materials Division in Gaithersburg, MD are presented. The CMML infrastructure was developed with funding from NIST and the NSF Center for Advanced Cement-Based Materials (ACBM). The primary objective of the modeling laboratory is to provide a centrally located facility for the archive, and the exchange and dissemination of cement and concrete research information.

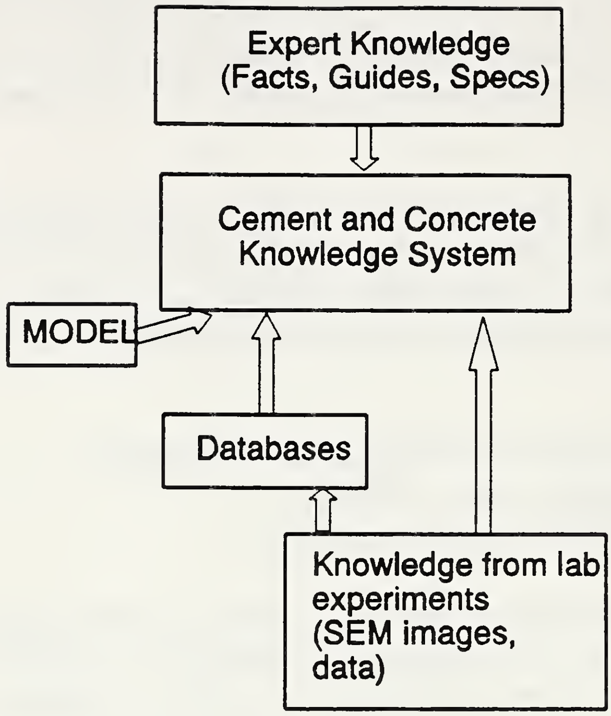

Figure 1. Knowledge system for cement and concrete.

2. CEMENTITIOUS MATERIALS MODELING LABORATORY

2.1 Benefits of a Centralized Modeling Laboratory

A centralized modeling laboratory for cementitious materials research has many benefits. It can:

[1] preserve the knowledge from the cement and concrete research area
[2] provide guidance to researchers on how to develop computer based models

[3] serve as a vehicle for the exchange of information (models, ideas, etc.)
[4] provide an infrastructure for the development of models
[5] influence the integration of other knowledge forms (expert systems, databases).

One of the major accomplishments of the CMML has been the establishment of an infrastructure that provides a modeling environment so that these potential benefits can be realized. A first step in the implementing the laboratory was to document, archive, and disseminate a model, using the Cement Hydration Simulation Model [Jennings 86] as a demonstration. It has been distributed to several researchers and remotely accessed on NIST host computers, and a reference manual has been written that describes the model's technical capabilities, and how to use it [Struble 89]. It is hoped that this effort will stimulate additional modeling efforts to be archived in the CMML.

Another major activity of the CMML is the sponsorship of a workshop on computer based modeling. In 1990, the first workshop was held at NIST. It was successful in stimulating interest in computer modeling for graduate students, postdoctoral fellows and faculty of the ACBM. The success of this activity has resulted in plans for a future workshop to be held in 1991.

2.2 Objectives of the Cementitious Materials Modeling Laboratory

Originally, it was thought that developing an infrastructure consisting of host computers, network and staff consultants would promote the development of models. In reality, relatively little computer modeling was at first being conducted outside of NIST. Also, modelers at other ACBM institutions used different computer resources, a difference that constrained the access and use of the CMML at NIST. In 1990, refinements were made to the CMML objectives to reflect the real-world status of cementitious materials modeling. These objectives should provide an incentive for modelers to submit their models for archive and promote consistent methods for model development. The revised objectives are to:

[1] develop guidelines and methods for establishing models
[2] conduct workshops on modeling methods
[3] provide access to NIST host computers (guest workers, networks)
[4] expand the scope of the laboratory to include other ACBM and other cementitious modelers

Also, an advisory committee consisting of a member from each of the ACBM participating organizations was established. The role of the advisory committee is to promote the development, archive and exchange of computer models for the CMML.

2.3 Cementitious Materials Modeling Laboratory Resources

Four fundamental components are needed to develop and maintain a centralized modeling laboratory: 1) computer hardware and software systems, 2) communications and networking equipment, 3) advisory and consulting staff, and 4) guidelines for operating the facility. The CMML has succeeded in establishing the first 3 components; this paper is an initial attempt to address the main issues for the 4th. It will be necessary to modify and refine these components as the facility gains more experience and more models are archived.

Efforts to build a modeling environment for the CMML began as early as 1985 with the super-minicomputer and graphics equipment used for the development of the Cement Hydration Simulation Model. Since then, resources have been added to allow participants of the ACBM and other organizations to utilize the laboratory. Researchers who use the CMML now have a number of resources available (described below):

[1] host computers (Convex C-120, Concurrent 3240, Cray II[1])
[2] high-speed color imaging computer (Sun/Pixar)
[3] electronic remote bulletin board computer
[4] local and wide-area networks
[5] dial-up telecommunications
[6] archived models and programs
[7] published papers on modeling topics
[8] modeling newsletter

NIST Host Computers. A Convex C-120 and a Concurrent 3240 computer system are installed at NIST in the Building and Fire Research Laboratory. These computers are available to modelers in a remote access capacity and for guest worker visits. A Cray II supercomputer is installed at NIST and managed by the NIST Central Computer Services Division. This facility is available for guest worker visits only.

High-speed Color Imaging Computer. A Sun workstation host to a Pixar II color imaging computer is used to develop mathematical models using computer imaging for visualization. This computer is available for workshop attendees, and guest workers and can be accessed for file transfer operations from local and wide-area networks.

---

1 This and other trade names and company products are identified to specify adequately the capabilities and procedures for the CMML. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

Remote Bulletin Board System (RBBS) Computer. Access to a microcomputer via dial-up telecommunications is provided for the exchange of information between users (files and messages) and to allow researchers to obtain information on cement and research topics. A NIST document [Kaetzel 90] has been written that describes the RBBS capabilities and how to use it.

Local and Wide-Area Networks. Networks are essential to move information, electronically, between researchers, and to provide access to remote computing facilities. The NIST Ethernet local area network is connected to the CMML. This allows remote access via wide-area networks such as BITNET, ARPANET, NSFNET, and others. Networking facilities are described later in this paper.

Dial-up Telecommunications. Although this type of communication is used less frequently today, it is a simple method for communicating information. The only CMML resource that uses this method is the RBBS. A disadvantage of this method is the cost for long distance telephone calls. Past experience with the CMML suggests that for this reason, researchers are reluctant to use this facility on a frequent basis.

Archived Models and Programs. This is a major objective of the CMML. By increasing the number of available models and programs in the laboratory, the benefits will be realized. The currently available models in archive are:

> [1] Cement Hydration Simulation Model - simulates the microstructure that develops during the reaction with water of portland cement, or more precisely, of its principal constituent tricalcium silicate (C_3S).
>
> [2] Cement Hydration Model - predicts the hydration kinetics of portland cement, including phase fractions, heat of reaction and the concentration of dissolved species.

Published Papers On Modeling Topics. The most notable published paper that relates to the subject of this paper is the "Manual for the Cement Hydration Simulation Model" [Struble 89]. This paper documents the capabilities and use of the Cement Hydration Simulation Model. It serves as an example of adequate documentation for archived models in the CMML.

Modeling Newsletter. A modeling laboratory newsletter is published periodically. The newsletter focuses on cementitious materials research activities.

3. SOFTWARE LANGUAGE DESIGN CONSIDERATIONS

The importance of developing understandable computer programs becomes obvious when one is faced with the task of enhancing or correcting programs that were written by another programmer several years earlier. Ideally, the program contains sufficient comments and standard syntax to assure readability. In real life however, this is often not the case. Simple programs grow to large complex programming systems that are rarely documented. This is due in part to the nature of scientific programming. The developer is in a research mode, and changes are frequently necessary to accurately represent the situation being modeled. However, there are many features that can make the development and maintainability of computer programs easier and reduce the need to "throw away" whole systems and start over.

It is recognized that developing mathematical models is significantly different than business applications. The steps in developing business applications can be more clearly defined and often require the resources of several programmers. Conversely, mathematical models are typically developed by one or two researchers/programmers who work together in a dynamic state toward a solution. Recommended steps in developing models are outlined in Figure 2. This paper focuses only on model design, program design, coding, testing, and documentation. These steps are addressed most often by mathematical modelers, and they are therefore the most important steps in developing mathematical models.

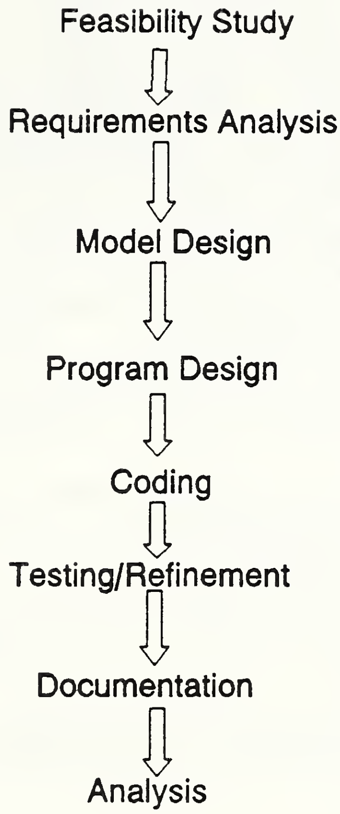

Figure 2. Steps in implementing computer based models.

3.1 Software Engineering Guidelines

Developing computer programs for mathematical models for the CMML must include considerations for maintainability, transportability, and accessibility. The attributes for these are described below.

Maintainability is a characteristic that enables modifications and enhancements to be made more easily. The characteristics of maintainability are modular program structure and readable programming style:

Modular Program Structure. Modular program structure is the separation of computer programs into logically discrete elements. Figure 3 illustrates an example of a modular programming configuration. Modular structure is necessary to separate computational program modules from input and output modules such as data preparation programs and graphics programs. The benefit of this feature is to allow program modifications without adversely affecting other programs in the model. Often it is necessary to make a change to a model for a single operation (e.g. formula) or to interface a new graphics library. Modularity allows the programmer to efficiently locate the affected program(s), make the change, and test the modification. Since many different programs (possibly from many different sources) are typically used in developing a model, as illustrated in Figure 4, a modular program structure also allows a significant reduction in the time required to regenerate an executable version of the model.

Readability. This characteristic allows a programmer to read and understand the statement syntax for a program module. To provide readability, a programmer must:

[1] assign names to variables that reflect their meaning
[2] include comments, when necessary in the program to draw attention to an operation
[3] identify the device assignments by explaining their function
[4] include revision numbers, dates and author's name in program comments
[5] include a brief comment to describe the purpose of the program
[6] provide a list of the input and output data elements necessary to use the program

Figure 5 is a simple program written in FORTRAN that illustrates these features.

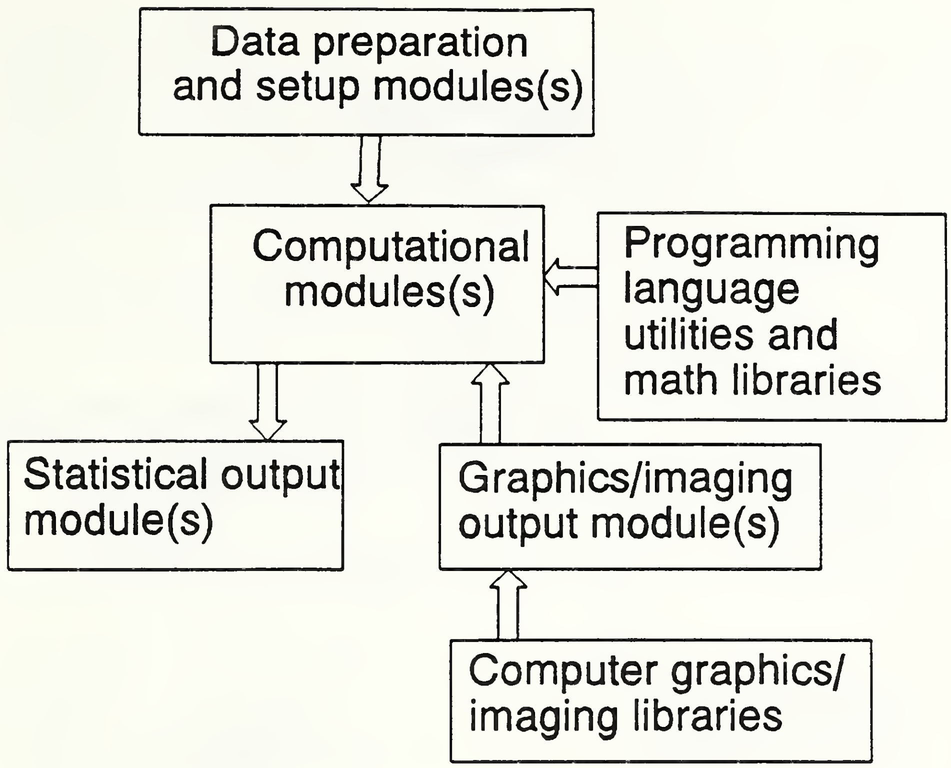

Figure 3. Example of a modular programming configuratioı

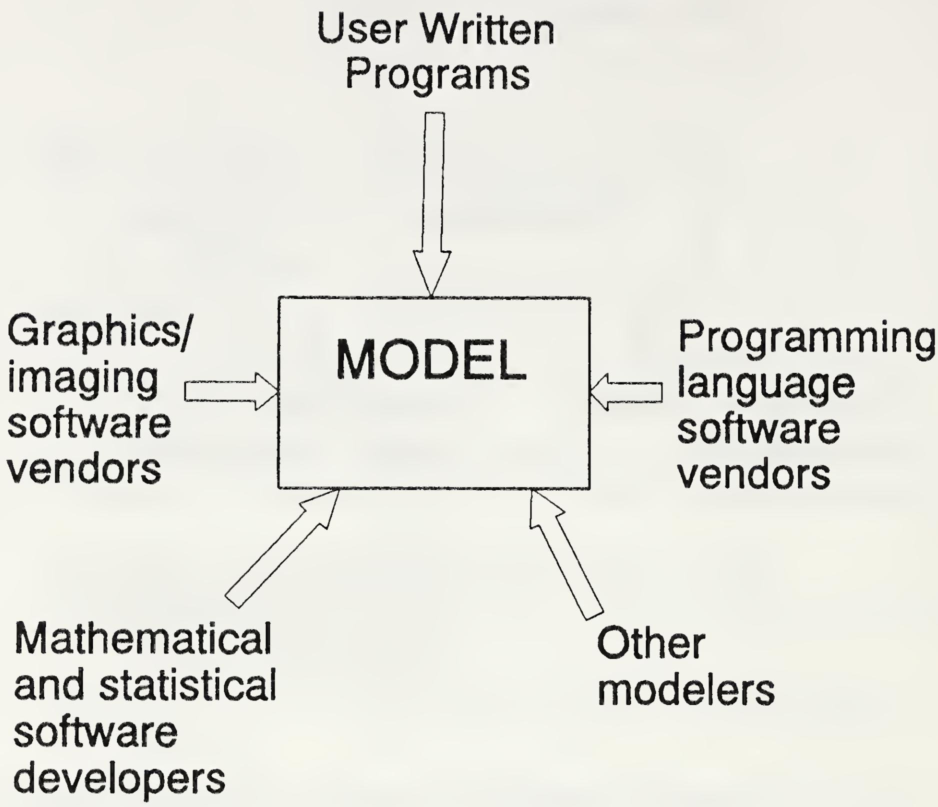

Figure 4. Sources of programs for developing computer models.

```
c       Program FASGRF
c       Written by Lawrence Kaetzel - CBT - May 24, 1984, Version 1.0
c       This program is designed to read 2 data files containing
c       1,000 values of x coordinate and y coordinate data and
c       calls a graphics sub-program that draws the curve on the
c       screen.
c
c       Logical unit assignments:       Unit 6= output screen
c                                       Unit 10=file containing x data
c                                       Unit 11=file containing y data
c       Variables:      xdta = array for x data
                        ydta = array for y data
                        n = number of pairs of data points
c
        dimension xdta(1000),ydta(1000)
c
c       Read x and y data
c
        do 100 i=1,1000
          read(10,*,iostat=istat) xdta(i)
            if (istat.ne.0) then
            write(6,*) ' Non-zero status from data file read: ',istat
            go to 150
            endif
          read(11,*,iostat=istat) ydta(i)
            if (istat.ne.0) then
            write(6,*) ' Non-zero status from data file read: ',istat
            go to 150
            endif
        100  continue
c
c       Call the graphing program
c
        150 n=i-1
        call graf(xdta,ydta,n)
        stop
        end
```

Figure 5. Sample program written in FORTRAN to illustrate readability features.

Transportability is a characteristic that allows programs to be moved between different computers. Since computers of different manufacturers maintain significantly different operating environments, it is sometimes difficult to design computer programs that are easy to transport. However, the following are minimal design features that will provide easier portability:

[1] Select a standard computer programming language (e.g. FORTRAN 77) to develop the model.
[2] If the model is to be transported between mainframe and microcomputers, consider any memory capacity constraints that may exist on the microcomputer, the number of bits used to represent a single and double-precision word, and the storage capacity for data files.
[3] Design computer code to utilize memory swapping and overlay features that may exist on the smaller computer.
[4] Establish data file record lengths that are compatible with the file system on the the smaller computer.
[5] Use standard language conventions for file manipulation (e.g. FORTRAN language conventions OPEN, CLOSE, READ, WRITE, INQUIRE).

Ideally, transporting programs between computers involves the following steps:

[1] Move the source program and test files to the destination computer.
[2] Identify the parallel device assignments for the destination computer.
[3] Re-compile the program and link the new program libraries.
[4] Verify the model using test data.

Real-world situations reveal that programmers will use language extensions specific to a computer manufacturer to improve performance. This requires replacement of computer code to incorporate the parallel feature found on the destination computer. Input/output functions and math library functions are examples of language extensions that differ among different computer manufacturers.

Accessibility is important in the design and use of a mathematical model to provide easy access and a smooth running development and production environment. Models can be accessed and executed using many different methods. Examples of these methods include:

[1] host computer batch and interactive processing accessed through remote dial-up and networks
[2] host computer computational batch with local interactive graphics display
[3] Single-user engineering workstation with interactive processing

Factors such as interactive response from the host computer, the availability of adequate development tools (e.g. language compilers, graphics libraries) on single user workstations, and costs involving remote access can dramatically affect the desirability of each computer system. Table 1 illustrates the advantages (annotated with a "+"), and disadvantages (annotated with a "-") of computer systems used to develop mathematical models. Assuming that the most efficient computer platform has been chosen, easy access must be provided to stimulate use. This subject is discussed in detail in a later section of this paper.

COMPUTER TYPE	MEMORY	SUPPORT SOFTWARE	CONSULTING STAFF	LANGUAGES	GRAPHICS RESPONSE	CONNECTIVITY
MAINFRAME	+ large capacity with best memory management	+ good graphics, math libraries and scripts	+ good support to users, but system not application oriented	+ most mature FORTRAN language for scientific use	- poor	+ best
MINI	+ may be restrictive due to established priorities	+ good graphics and math libraries	+ best and focused support for applications	+ FORTRAN and C well developed, others less mature	+- fair to good	+- may be limited
SCIENTIFIC/ ENGINEERING WORKSTATION	- may be limited	+ good graphics	- user normally relies on vendors	+ FORTRAN and C best choices	+ best	+ good
PERSONAL MICRO	- may have constraints due to operating system limitations	- limited graphics and math libraries, quality an issue	- user must rely on vendors	- FORTRAN less developed; good C languages	-+ best, but resolution limited	+ good

Table 1. Modeling computer system strengths and weaknesses.

3.2 Language Selection

The selection and use of a computer language for developing a computer model is often dictated by the "most convenient" tool that is already available to the developer. Convenience is provided by one or more of the following circumstances:

[1] the availability of a programmer who is familiar with a specific language
[2] the availability of a language compiler or interpreter on a local computer
[3] the language used is represented in a previous version of the model.

Seldom is the development language selected on its performance or syntactical merits. Such selection may require an extensive evaluation or comparison of the resources of different manufacturers. In the absence of a central consulting group, this can become a very time consuming and frustrating exercise. One benefit of a central computing resource such as the CMML is that much of the work has been done in selecting computer platforms and language processors and recommendations are already made concerning their strengths and weaknesses.

Most computer based models for cementitious materials use either FORTRAN or C language processors. FORTRAN has been criticized for its unstructured nature and lack of development aids to debug programs. Yet it is still the most common language used in the development of computer programs for scientific use. This is because of its maturity, standards activities [ANSI 78] that promote the transportability of the language,and the large base of knowledgeable programmers. The C programming language is rapidly growing in use, especially on small computer workstations. The ANSI Committee X3J11 has developed a standard for the C programming language [ANSI 89], and many references are available on the subject [Jaeschke 88, Kochran 88].

One of the significant improvements that can result in using the C programming language is the available support utilities. These include; graphics programs, debugging aids, and programs for memory management. They are developed by the software vendors to take advantage of the software operating system and computer hardware architectures. FORTRAN and other languages also provide these capabilities, but they are not as robust as C.

In selecting a programming language for developing a model, one must consider the development and run-time features supported by the language, its accuracy and precision, and its transportability. Table 2 presents a comparison of the FORTRAN and C programming languages. The CMML has both languages available on its computers. Further development of computer based models in the CMML will focus on visualization, and significant factor in selecting a language will be how well the language facilitates visualizing results through improved computer graphics capabilities and color imaging. Other factors include the need to adapt models to different computer platforms and the integration of models into knowledge based systems.

FORTRAN LANGUAGE	C LANGUAGE
strong capabilities for complex numbers	good memory management
long development time compared to C	shorter development time compared to FORTRAN
language standard exists	language standard exists
limited support libraries	robust support libraries
large body of programmers	growing and more diverse programmers
development aids limited	best set of aids for development

Table 2. FORTRAN and C language comparison.

4. DOCUMENTATION

Good documentation is an essential part of developing a computer model, especially if it is to be used by others. It is seldom developed adequately to describe the technical aspects as well as the operational procedures. The reasons are obvious: good documentation takes time, and revisions are frequent. Normally, computer documentation contains a large amount of jargon, may be inaccurate (due to precision needed to explain operating commands), and is outdated quickly. Given that these problems exist, there are steps that can be taken to improve the readability, accuracy, and completeness of computer model documentation. Perhaps the most important aspect is to write a document that addresses the intended audience. A technique for development good documentation can be found in published books by Sides [Sides 84] and Grimm [Grimm 82].

Two NIST reports have been recently published on computer based models [Struble 89 and Jones 90]. The first [Struble 89] is a manual describing the technical capabilities and use of the Cement Hydration Model, which was archived in the CMML. The report contained the following sections on the model:

[1] Introduction
[2] A description of the area of research
[3] Technical description of the model (background, overview and implementation)
[4] Description of work to produce current version
[5] References of related papers
[6] Description of computer programs representing the model
[7] Flow diagrams for the modeling processes
[8] List of program variables
[9] Data file structures

[10] Sample Sessions
[11] Test data
[12] History of revisions
[13] Description of development computer
[14] Computer user's guide
[15] Guidelines for access
[16] Guidelines for future revisions
[17] Examples of output (statistics, graphics)

Recognizing the need to keep the document current, items 7-17 were developed as appendices for the manual which can be rapidly changed. Use of appendices is good practice because it provides an easy way of disseminating new information to users, and it eliminates the need to publish a new manual in its entirety.

5. SELECTION OF COMPUTER PLATFORMS

The availability of low-cost microcomputers has made the acquisition of computer platforms for computer based model development easy but not necessarily the most desirable. Often the platform is selected without adequate consideration for performance, quality, and development time. Several important factors in the selection of computer hardware and software must be considered if a model is to meet its objectives. These factors are summarized below.

Memory Size. Memory requirements for a model may be obvious from the start. Calculations for estimating the size of matrices for example, can be made to determine the memory requirements for representing arrays of values or strings. What is critical however, is the potential for growth for the model. Array sizes tend to grow exponentially. These requirements in combination with the need to have mathematical libraries, graphics libraries, and support programs can quickly saturate a small computer system's memory capacity. Mid-sized and mainframe computers have operating environments that utilize a swapping or overlay feature that allows an application to consume more memory than is allocated to the task. Control of these operations may also be available to the programmer when the model is being developed. An example of an overlay design technique for structuring program modules is illustrated in Figure 6. This feature is available on many computer platforms, especially microcomputers. The use of this feature also enhances program readability and comprehension.

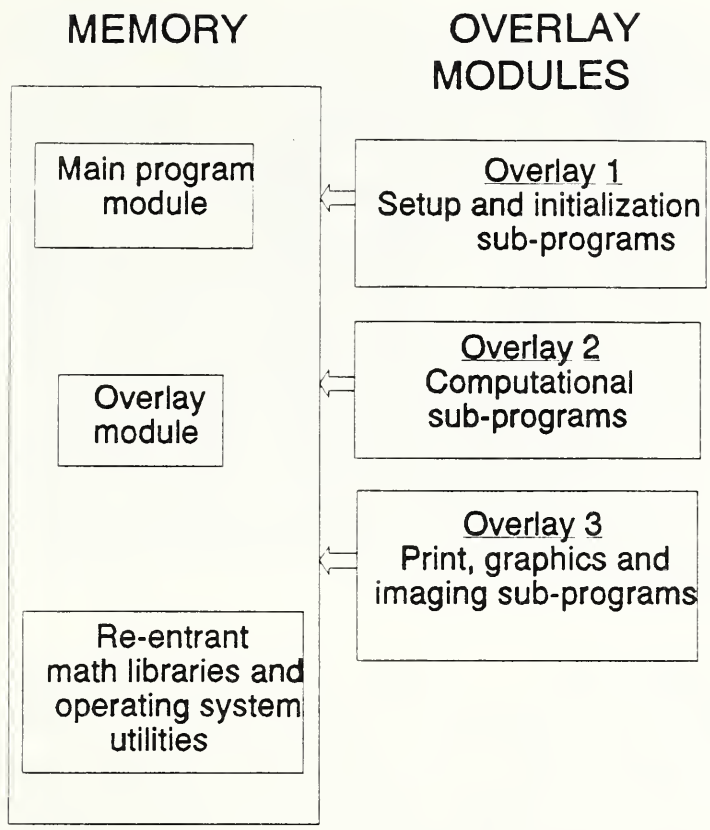

Figure 6. Example of program overlay design.

It is difficult to predict exactly the amount of memory a model will require. Constraints that may occur later in the development can be reduced or eliminated by asking the following questions.

[1] Does the computer support a robust memory management system (virtual, swapping)?
[2] Does the language processor or operating system restrict the size of an application (e.g., DOS based microcomputers)?
[3] Does the operating system support re-entrant (re-usable) code for libraries?

If the answer is no, then it is important to make very accurate predictions on memory size and possibly select an alternative computer platform.

Auxiliary Storage. This type of storage, sometimes referred to as memory, can have a dramatic effect on the performance of the computer system. Devices such as disk, additional memory, and magnetic tape are used to store computer programs and data. Important factors in estimating adequate resources are; speed of the device and the amount of storage. On mid-sized and mainframe computers this is decided by the computer manager. Cost however, may be an issue. There may be a charge for keeping computer files resident on the storage unit. For smaller computers such as scientific and engineering workstations, and desktop microcomputers, the task of estimating the needed capacity, and providing data security is left with the end-user. Historically, the greater the capacity of disk, for example 300-1000 mega bytes and the fastest disk possible will be needed. One-quarter inch tape drive systems are commonly used to backup files that are stored on disk. These devices can store as much as 250 million characters of information on a single tape.

Computational Speed. Speed is the most advertized and usually the most talked about attribute of a computer system. Speed means different things to different users. For the purpose of the paper, speed is defined as the time required to compute an arithmetic value by the central processing unit. Standard benchmarks exist and can be obtained easily. The most common are Whetstone and Drystone benchmarks. These are relatively good measures of a computer's speed for calculations. The graph in Figure 7 shows some relative speeds of computers in the CMML. Computational speed should not be the final deciding factor in choosing a computer based on performance issues. Earlier in this paper, other significant factors such as; interactive response, development time, and access to the computer were equally, if not more important. The misconception associated with microcomputers is that "oh well, I can just start the computer program and come back tomorrow morning and get my results". This sounds reasonable at first, but in real life it may present problems for several reasons. First, the computer is often not available for other use while the model is running. Or, the model may abort and result in a loss of information, hence another overnight run. Alternative computers, such as higher-performance microcomputers or workstations that allow multiple programs to execute concurrently may be more suitable.

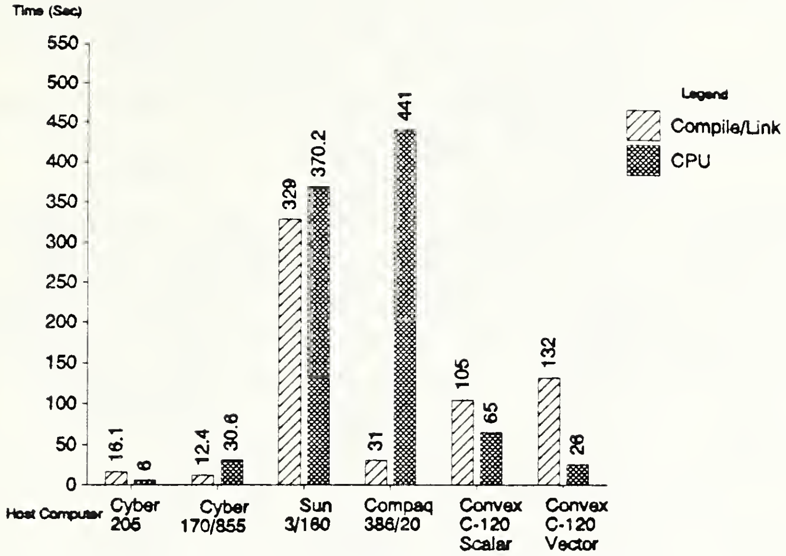

Figure 7. Timing benchmarks for CMML computers.

Software Development Languages and Tools. Computer systems support varies widely between mainframe and microcomputers. At the mainframe level, a large support staff is often available to develop operating procedures or scripts, programs, documentation and provide consultation on-line. At the lowest level, it is often you versus the computer. Seldom can you get through to a software or hardware vendor (especially in the area of scientific computing) without a long wait or being transferred to different groups. Problems that arise from using scientific tools are often complex and can not be addressed by first-line support. Retailers and distributors can seldom answer questions. This reveals two weaknesses in selecting smaller computer based tools; 1) you may have to resolve the problem yourself or find a way around it, or 2) the program may run well but it gives wrong answers (which may be discovered much later). It is nearly impossible to eliminate all problems arising from the use of hardware and software. It is however, possible to reduce them by applying strong criteria when evaluating and purchasing them. Some general rules of thumb are:

[1] Evaluate computer products by running tests and benchmarks.
[2] Choose standard or widely used language processors and libraries.
[3] Don't sacrifice quality for price.
[4] If strong support is essential, choose a centrally supported mainframe or minicomputer facility.

[5] Choose a computer architecture that has at least 32 bit word lengths for representing real and integer numbers (smaller architectures may have less precision).
[6] Don't rely solely on advertisement. Consult other users with similar applications.

Communications Capabilities. In the past, communications normally was referred to as terminal to host connectivity. Today, however, the definition is "intelligent terminal/processor" to host. It is nearly impossible to be productive without a networked computing environment that connects workstation to workstation, workstation to host, and host to host. The advances in networking interconnectivity and standard protocols for information exchange have allowed this to happen. It is now possible to connect computers of different manufacturers, different operating systems, and different formats, together for terminal emulation, file transfer, and message switching. Choosing a computer that supports strong networking facilities enhances the development and transfer of computer models. Techniques for accomplishing this connectivity are described later in this paper.

Interpretation of results. Computer graphics and imaging are state-of-the-art methods for viewing the results of computer based models. These capabilities offer first-hand, visual interpretations of simulations, in real-time. They are important in modeling because they provide an ability to assess the progress, view phoneme, and condense the output into a manageable form. Most models developed today use 2D and 3D representations of objects such as cement particles and their constituents. At NIST, computer imaging has been used to develop models for cement and concrete [Bentz 90]. This technique provides enhancements compared to large quantities of statistical data that is often difficult to comprehend. Computing capabilities that use visualization features (e.g. graphics, and imaging) are configured as distributed systems. A large mainframe or mini provides the compute capabilities, augmented by a remote workstation that supports program development and visualization. Figure 8 illustrates the existing architecture of the CMML's distributed processing environment.

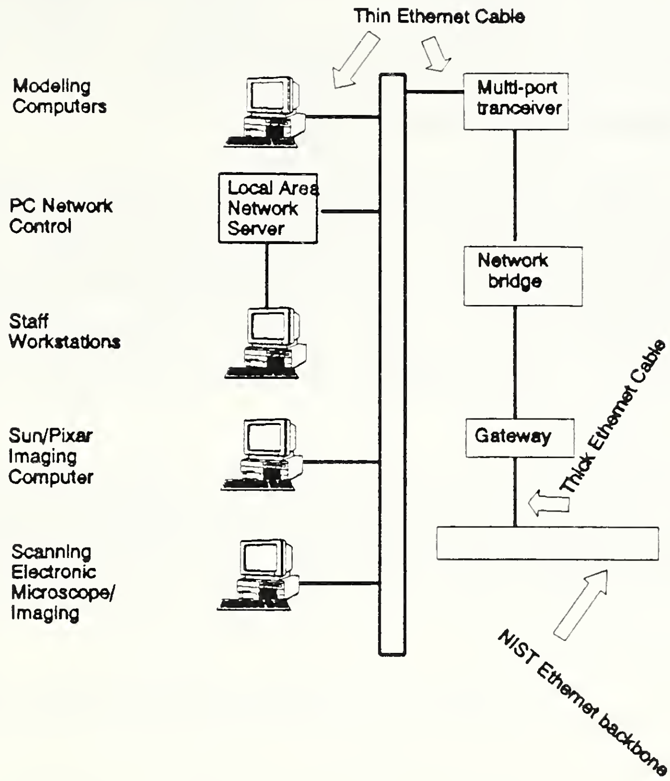

Figure 8. CMML's distributed processing environment.

6. ACCESSIBILITY

Access to computer models and development resources can be made using wide-area and local area networks, direct on-site connections, dial-up telecommunications or through the transfer of computer media such as magnetic tapes and disks. Clearly, the most efficient is the use of computer networks. Computer networks have become popular among scientific organizations world-wide. The Ethernet[2] protocol is the primary method of information exchange in the CMML. As facilities mature at all levels of computing and at other organizations, networking will become the sole method of communication. The main reasons for success in the use of networks is:

[2] Ethernet is a trademark of Xerox Corporation. It is a network protocol which is commonly used in the business and scientific community for the exchange of messages, and files, and for terminal emulation.

[1] the development of hardware and software standards
for information transfer
[2] the speed of information transfer (typically 10-50 mega
bytes per second)
[3] the installation of world-wide gateways
[4] the literacy of computing among all users
[5] the reduced cost of communication

Alternative methods to networking involve extensive delays in transferring media, and expensive telephone charges for dial-up telecommunications. Dial-up telecommunications constrain the exchange of information and therefore does not promote the concept computer modeling. One system in the CMML uses the dial-up method to provide bulletin board capabilities, but this will be replaced in the future as more advanced software becomes available.

Procedures for using networks and dial-up telecommunications vary greatly. The requirements and capabilities are presented in the remainder of this section. Table 3 shows the various CMML computing resources and their access methods.

Computer	LAN Method	Dial-up Method	On-site	Service Provided
NIST Cray II	✓	✓		1. model development 2. computational 3. Ethernet network services
BFRL Convex	✓			1. model development 2. computational
BFRL Concurrent		✓	✓	1. computational 2. graphics
CMML Sun/Pixar imaging computer	✓		✓	1. image analysis 2. computational 3. model development
CMML Staff workstations	✓			1. networked workstation 2. model development
Remote Bulletin Board		✓		1. information dissemination

Table 3. CMML computer access methods.

6.1 Dial-up Telecommunications

This form of connecting terminals and computers to host computers has been in existence since the early days of computing. The communications equipment used today is virtually the same, except for the use of digital circuits instead of analog, speeds have increased by roughly a factor of ten (from 110 characters per second to 19.2 thousand characters per second), and the information format is still primarily ASCII. This method is also potentially the most expensive to the individual user. In most cases, the user must pay for connect time on the computer, and the time for using the telephone circuit (for long-distance calls). Historically, this has reduced the use of remote facilities to the CMML, even though there are not charges for computer time. A schematic illustrating the dial-up connection for a remote user to the CMML computers is shown in Figure 9. Hardware requirements are a computer terminal that supports the ASCII code, or a computer workstation that has the capability of emulating a computer terminal, and a modem connected to a commercial telephone exchange. One advantage of using this method is that the operating commands are quite simple compared to networking procedures that are generally more complex until a user becomes familiar with a sub-set of the commands and develops a customized set of procedures.

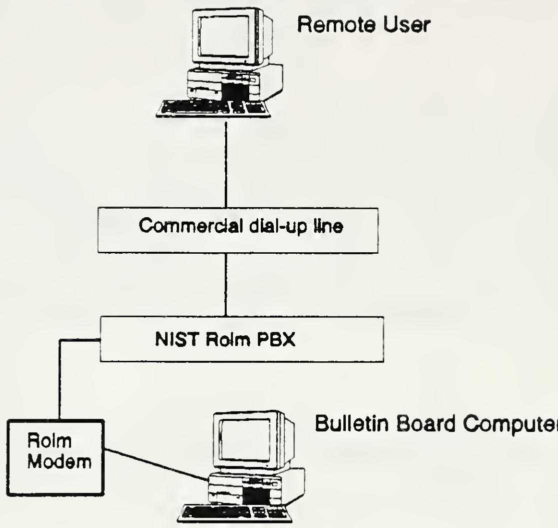

Figure 9. Illustration of dial-up access method.

6.2 Network Communications

Networks are very complex systems of communications protocols, gateways (host communications processors), and cables. The fundamental advantage to the user is that most networks are controlled and maintained by a support staff at an organization. This relieves the user of the responsibility of installing and maintaining the network hardware and software. Typically, the user will obtain the necessary printed circuit board (network link board), workstation software, and arrange to have a cable installed from the network bus interface to their work area. A simplified illustration of a Ethernet protocol network interface for a workstation is shown in Figure 10. The cost of connecting to the network in using this method is often between \$.5K and \$1K, including software. Once this expenditure is made, the cost of using the network is often underwritten by the organization. Few installations charge each user directly for connections to the network. Network charges for gateways and host computers are normally considered overhead or administrative costs.

The CMML has developed an infrastructure that extensively utilizes wide-area and local area network facilities world-wide. Users of the facility can access host computers and workstations locally (at NIST) and throughout the world. Figure 11 shows the topography of the NIST CMML connections. Through the connection labelled WAN, users can access such world-wide networks as BITNET, ARPANET, NSFNET, and JANET.

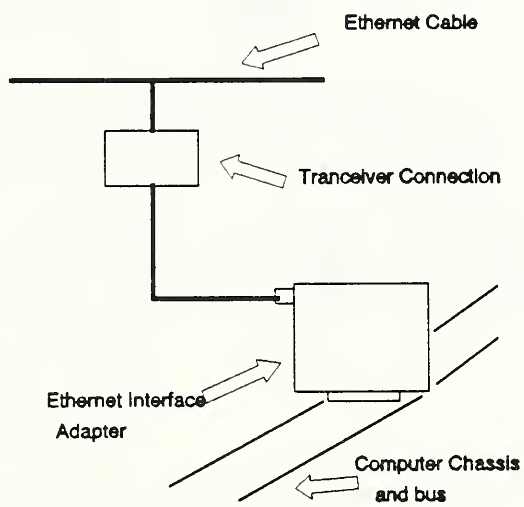

Figure 10. Ethernet LAN workstation interface.

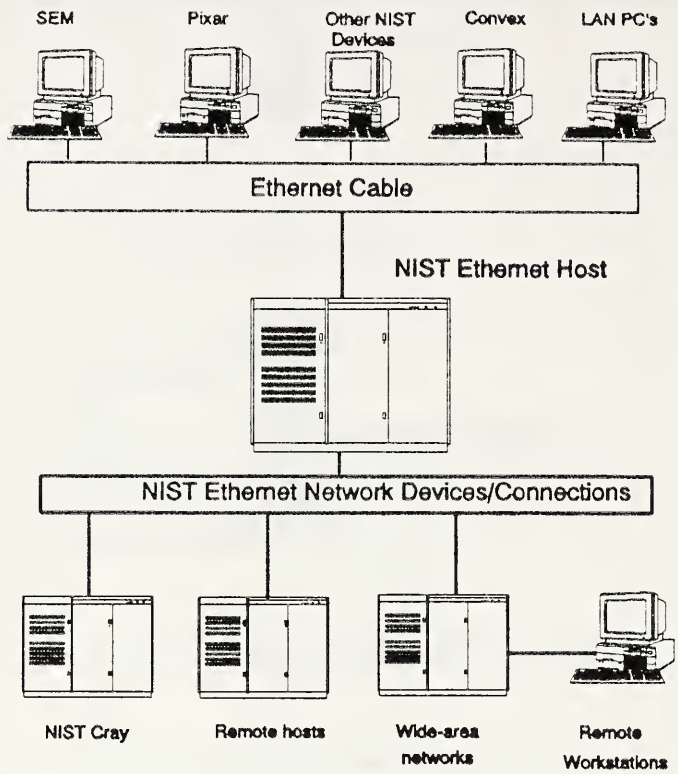

Figure 11. NIST Ethernet LAN topography.

7. ARCHIVE AND DISTRIBUTION OF MODELS

The archive of computer models in the CMML will preserve the knowledge contained in the models; distribution provides access to models for users who cannot or do not wish to access the facility remotely. Archiving the models in computer readable form and in a consistent way, using the guidelines presented in this paper will promote their exchange. Growth in the number of models archived in the facility will require the development of additional capabilities to allow more efficient retrieval and distribution of the models.

Interested researchers may request copies of the models held in archive by sending a request to the CMML staff. Responsibility for the accuracy of the model, its support and documentation is that of the author. Models are chosen for archive by the CMML Advisory committee. The committee evaluates the usefulness of the model for wide-spread distribution and its potential for advancing the science of cementitious materials.

8. CONCLUSIONS

This paper presents the concept of a centralized facility for the development, archival and distribution of computer based models for cementitious materials research. It provides information to current and potential modelers on how to evaluate the computing resources needed to develop models, how to design models, how to document models and how to use the resources of the CMML. The results of this effort will cause more and higher quality models to be developed in a consistent manner. The concept is important in the development of more complex knowledge systems that include expert systems, databases, and computer based models.

9. ACKNOWLEDGEMENT

The activities described in this paper were funded by the National Science Foundation Center for Advanced Cement Based Materials. NIST is a cooperating organization along with Northwestern University, University of Illinois, University of Michigan, and Purdue University. The CMML infrastructure development was supported by NIST. Also, the authors wish to recognize Dr. Geoffrey Frohnsdorff, Chief, NIST Building Materials Division for his guidance and encouragement in establishing the CMML.

10. REFERENCES

American National Standards Institute [ANSI], "American National Standard Programming Language FORTRAN", ANSI X3.9-1978, New York, NY [1978].

American National Standards Institute [ANSI], "American National Standard Programming C", ANSI X3.159-89, New York, NY [1989].

Bentz, D.P., Garboczi, E.J., "User's Guide to CMMAP: Cement Microstructure Modelling and Analysis Package". NIST Internal Report 90-4442, Gaithersburg, MD [1990].

Bentz, D.P., Garboczi, E.J., "A Digitized Simulation Model for Microstructural Development", Proceedings of the NIST/American Ceramic Society Conference on Advances in Cement-Based Materials, Gaithersburg, MD [1990]

Cross, M., Gibson, R.D., O'Carroll, M.J., Wilkinson, T.S., "Modelling and Simulation in Practice", John Wiley & Sons, New York, NY [1979].

Grimm, S.J., "How to Write Computer Manuals for Users", Lifetime Learning Publications, Belmont, CA [1982].

Jaeschke, R., "Portability and the C Language", Hayden Books, Indianapolis, IN [1988].

Jennings, H.M., Johnson, S.K., "Simulation of Microstructure Development During the Hydration of a Cement Compound", Journal of the American Ceramic Society, 69(11) pp. 790-795 [1986].

Jones, W.W., Forney, G.P., "A Programmer's Reference Manual for CFAST, the Unified Model of Fire Growth and Smoke Transport", NIST Technical Note 1283, Gaithersburg, MD [1990].

Kaetzel, L.J., Clifton, J.R., "Guide to the Use of the Cement and Concrete Research Remote Bulletin Board System (RBBS) Computer", NIST Internal Report 4473, Gaithersburg, MD [1990].

Kochran, S.G., "Programming in ANSI C", Hayden Books, Indianapolis, IN [1988].

Sides, C.H., "How to Write Papers and Reports About Computer Technology", ISI Press, Philadelphia, PA [1984].

Spriet, J.A., Vansteenkiste, G.C., "Computer-aided Modelling and Simulation", Academic Press, New York, NY [1982].

Struble, L.J., Johnson, S.K., Hartmann, M., Kaetzel, L.J., Jennings, H.M., "Manual for the Cement Hydration Simulation Model", NIST Technical Note 1269, Gaithersburg, MD [1989].

NIST-114A
(REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4650

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
August 1991

4. TITLE AND SUBTITLE

Guidelines for the Development of Computer Based Models ina Cementitious Materials Modeling Laboratory

5. AUTHOR(S)

Lawrence Kaetzel, James Clifton, Leslie Struble

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This paper presents guidelines and considerations for the development, archiving, and distribution of computer models for a centralized cementitious materials modeling laboratory. An analysis of the approach used by cement researchers to develop large complex computer models reveals the need for guidelines in selecting computer platforms, software languages and tools, software engineering and documentation. A modeling laboratory established at NIST associated with the NSF Center for Science and Technology of Advanced Cement-Based Materials is discussed and the important of such a facility in promoting the exchange of information (i.e. ideas, models, data). The computer models currently archived in the modeling laboratory are use as test cases to describe the facility.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

13. AVAILABILITY

- [X] UNLIMITED
- [ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
- [ ] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
- [X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
34

15. PRICE
A03

ELECTRONIC FORM